AD-A049 425 WISCONSIN UNIV MADISON MATHEMATICS RESEARCH CENTER F/G 12/1
A VARIABLE METRIC METHOD FOR LINEARLY CONSTRAINED MINIMIZATION --ETC(U)
OCT 77 K RITTER DAAG29-75-C-0024
UNCLASSIFIED MRC-TSR-1798 NL

| OF |
AD A049425

END
DATE FILMED
3-78
DDC

(13)

MRC Technical Summary Report # 1798

A VARIABLE METRIC METHOD FOR LINEARLY CONSTRAINED MINIMIZATION PROBLEMS

Klaus Ritter

Mathematics Research Center
University of Wisconsin—Madison
610 Walnut Street
Madison, Wisconsin 53706

see back page for 1473

October 1977

(Received August 1, 1977)

Sponsored by

U. S. Army Research Office
P. O. Box 12211
Research Triangle Park
North Carolina 27709

ACCESSION for		
NTIS	White Section	X
DDC	Buff Section	☐
UNANNOUNCED		☐
JUSTIFICATION		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

UNIVERSITY OF WISCONSIN - MADISON
MATHEMATICS RESEARCH CENTER

A VARIABLE METRIC METHOD FOR LINEARLY CONSTRAINED MINIMIZATION PROBLEMS

Klaus Ritter

Technical Summary Report #1798
October 1977

ABSTRACT

An algorithm is described for minimizing a nonlinear function subject to linear inequality constraints. The method generates a sequence $\{x_j\}$ with $x_{j+1} = x_j - \sigma_j s_j$, where s_j and σ_j denote the search direction and the step size, respectively. Associated with each x_j is an (n,n)-matrix $C_j = (c_{1j},\ldots,c_{nj})$ which is used to compute s_j as a suitable linear combination of $c_{1j},\ldots,c_{nj}$. At each iteration the matrix C_j is updated. The update formula depends on the constraints that are active at x_j and x_{j+1}, respectively. Under appropriate assumptions it is shown that $\sigma_j = 1$ for j sufficiently large and that $\{x_j\}$ converges superlinearly to the optimal solution of the minimization problem.

AMS (MOS) Subject Classification: 90C30

Key Words: linearly constrained minimization, variable metric method, superlinear convergence

Work Unit Number 5 (Mathematical Programming and Operations Research)

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

SIGNIFICANCE AND EXPLANATION

Linear programming deals with the problem of minimizing

$$c^T x$$

subject to

$$Ax \leq b, \quad x \geq 0 .$$

This has been spectacularly successful in certain classes of optimization problems that occur, for example, in management. However most situations are inherently nonlinear, and one would like to be able to deal with the general problem: minimize $f(x)$, subject to $g(x) \leq 0$, $x \geq 0$. Progress in developing practical and efficient methods for this full nonlinear problem is slow, as one might expect due to its generality. The present paper deals with an efficient computational method for solving the restricted problem: minimize $f(x)$, subject to $Ax \leq b$, $x \geq 0$, i.e. minimize a nonlinear objective function with linear inequality constraints.

The method depends on starting with a feasible solution x_0, i.e., a solution that satisfies the inequality constraints. Then produce a sequence $x_1, x_2, x_3, \ldots$ of feasible solutions by a variable metric method (see abstract). Under suitable assumptions these will converge superlinearly to the optimal solution.

The responsibility for the wording and views expressed in this descriptive summary lies with MRC, and not with the author of this report.

A VARIABLE METRIC METHOD FOR LINEARLY CONSTRAINED MINIMIZATION PROBLEMS

Klaus Ritter

1. Introduction

For many years variable metric methods have been used successfully in unconstrained minimization. In 1969 Goldfarb [7] extended the Davidon-Fletcher-Powell method [5] to problems with linear equality and inequality constraints. In [6] Gill and Murray described variable metric methods for linearly constrained problems which use approximations to the Hessian matrix of the objective function rather than to the inverse Hessian matrix. In all cases it has only been shown that the method determines the optimal solution in a finite number of steps if the objective function is convex and quadratic. Recently, Fischer [4] proved superlinear convergence of the Davidon-Fletcher-Powell and the Broyden-Fletcher-Goldfarb-Shanno [11] method for linearly constrained problems.

Brodlie, Gourlay and Greenstadt [2] and more recently Davidon [3] have investigated variable metric methods where the matrix which approximates the inverse Hessian of the objective function is factorized. In this paper such a factorized variable metric method for linearly constrained problems is given. Using Fischer's results [4] it is shown that it converges superlinearly. In the unconstrained case it reduces to a method which was investigated in [10] and shown to be a factorized version of the Broyden-Fletcher-Goldfarb-Shanno method.

Sponsored by the United States Army under Contract No. DAAG29-75-C-0024.

2. General description of the algorithm

We consider the following minimization problem: Minimize the function

$$F(x)$$

subject to the constraints

$$Ax \leq b ,$$

where $x \in E^n$, $b \in E^m$ and A is an (m,n)-matrix.

Throughout the paper it is assumed that the set

$$R = \{x | Ax \leq b\}$$

of feasible solutions in nonempty and that for every $x \in R$ the gradients of the constraints, active at x, are linearly independent. Furthermore, we assume that $F(x)$ is twice continuously differentiable and denote the gradient and the Hessian matrix of $F(x)$ at a point x_j by $g_j = \nabla F(x_j)$ and $G_j = G(x_j)$, respectively. In order to prove superlinear convergence we finally need the following

Assumption 1: There are positive numbers μ, η and L such that

$$\mu \|x\|^2 \leq x'G(y)x \leq \eta\|x\|^2 \quad \text{for all} \quad x,y \in E^n \tag{2.1}$$

and

$$\|G(x) - G(y)\| \leq L\|x - y\| \quad \text{for all} \quad x,y \in E^n .$$

Assumption (2.1) implies that $F(x)$ is uniformly convex and that there is a unique $z \in R$ such that

$$F(z) \leq F(x) \quad \text{for all} \quad x \in R .$$

Let $x_j \in R$ be a point determined by the algorithm. For ease of notation we assume that

$$a_i'x_j = (b)_i, \qquad i = 1,\ldots,q$$

$$a_i'x_j < (b)_i, \qquad i = q + 1,\ldots,m ,$$

where $a_1',\ldots,a_m'$ denote the rows of A. Set

$$T_j = \{x | a_i'x = 0, \quad i = 1,\ldots,q\}$$

and denote the orthogonal projection of g_j onto T_j by $\hat{g}_j$. Then g_j can be written as

$$g_j = \sum_{i=1}^{q} \lambda_i a_i + \hat{g}_j . \tag{2.2}$$

Because $F(x)$ is convex it follows from the Kuhn-Tucker-conditions (see e.g. [8]) that x_j is an optimal solution if and only if

$$\hat{g}_j = 0 \quad \text{and} \quad \lambda_i \leq 0, \qquad i = 1,\ldots,q .$$

Suppose x_j is not an optimal solution. Then we want to determine a search direction s_j and a step size $\sigma_j > 0$ such that

$$x_{j+1} = x_j - \sigma_j s_j \in R \quad \text{and} \quad F(x_{j+1}) < F(x_j) . \tag{2.3}$$

In order to guarantee that there exists an x_{j+1} with the properties (2.3) we need a search direction s_j with

$$g_j' s_j > 0 \quad \text{and} \quad a_i' s_j \geq 0, \qquad i = 1,\ldots,q . \tag{2.4}$$

If $\hat{g}_j \neq 0$ we can find an $s_j \in T_j$ with $g_j' s_j > 0$. In this case $a_i' s_j = 0$, $i = 1,\ldots,q$, and all constraints which are active at x_j are also active at x_{j+1}. If, say, $\lambda_q > 0$ we can determine an s_j such that $g_j' s_j > 0$, $a_q' s_j > 0$ and $a_i' s_j = 0$, $i = 1,\ldots,q-1$. In this case the constraint $a_q' x \leq (b)_q$ is not active at x_{j+1}.

It is well-known that in order to prevent zig-zagging the decision to drop an active constraint has to be made with some caution. Often it is based on a comparison between $\|\hat{g}_j\|$ and the maximal value of the multipliers $\lambda_1,\ldots,\lambda_q$, defined by (2.2). *We shall adopt the policy to choose* $s_j \in T_j$ unless

$$\|\hat{g}_j\| \leq \gamma_j \max\{\lambda_1,\ldots,\lambda_q\} . \tag{2.5}$$

Here $\{\gamma_j\}$ *is a convergent sequence of positive numbers with the property that*

$$\lim_{j \to \infty} \gamma_j = 0$$

if and only if $s_j \notin T_j$ for infinitely many j.

In order to compute $\hat{g}_j, \lambda_1,\ldots,\lambda_q$ and an s_j with the properties (2.4) we *associate with each* x_j, determined by the algorithm, a nonsingular (n,n)-matrix

$$C_j = (c_{1j},\ldots,c_{nj}) .$$

The columns $c_{q+1,j},\ldots,c_{nj}$ are chosen in such a way that they form a basis of the $(n-q)$-dimensional subspace T_j. For $i = 1,\ldots,q$ the vector c_{ij} is then uniquely determined by the equations

$$a_k'c_{ij} = 0, \quad k = 1,\ldots,q, \quad k \neq i$$

$$a_i'c_{ij} = 1$$

$$c_{kj}'c_{ij} = 0, \quad k = q+1,\ldots,n\ .$$

Because $c_{q+1,j},\ldots,c_{nj}$ form a basis of T_j it follows that the matrix

$$H_j = \sum_{i=q+1}^{n} c_{ij}c_{ij}'$$

is positive definite on the subspace T_j. Furthermore,

$$H_j x = 0 \text{ for every } x \in \operatorname{span}\{a_1,\ldots,a_q\}\ .$$

Multiplying (2.2) by c_{ij}' we have

$$\lambda_i = c_{ij}'g_j, \quad i = 1,\ldots,q\ .$$

Thus

$$\hat{g}_j = g_j - \sum_{i=1}^{q} (c_{ij}'g_j)a_i$$

and (2.5) becomes

(2.6) $$\|g_j - \sum_{i=1}^{q} (c_{ij}'g_j)a_i\| \leq \gamma_j \max\{c_{ij}'g_j,\ i = 1,\ldots,q\}\ .$$

If (2.6) is not satisfied we choose

$$s_j = H_j g_j$$

otherwise we set

$$s_j = c_{qj}$$

where $c_{qj}'g_j \geq c_{ij}'g_j$, $i = 1,\ldots,q$, say.

With s_j determined we can define the maximal step size σ_j^* as follows

$$\sigma_j^* = \min\left\{\frac{a_i'x_j - (b)_i}{a_i's_j}\ \middle|\ \text{for all } i \text{ with } a_i's_j < 0\right\},$$

where we set $\sigma_j^* = \infty$ if $a_i's_j \geq 0$ for all i. Following a method suggested by Powell [9] we compute a σ_j such that

$$F(x_j - \tilde{\sigma}_j s_j) \leq F(x_j) - \delta_1 \tilde{\sigma}_j g_j' s_j$$

and

$$(\nabla F(x_j - \tilde{\sigma}_j s_j))' s_j \leq \delta_2 g_j' s_j$$

with $\tilde{\sigma}_j = 1$ if possible. Here δ_1 and δ_2 are constants with $0 < \delta_1 < \delta_2 < 1$ and $\delta_1 < 0.5$. Finally we set

$$\sigma_j = \min\{\tilde{\sigma}_j, \overset{*}{\sigma}_j\} \quad \text{and} \quad x_{j+1} = x_j - \sigma_j s_j .$$

In order to complete a cycle of the algorithm we have to compute $C_{j+1} = (c_{1,j+1}, \ldots, c_{n,j+1})$. Depending on the constraints active at x_j and x_{j+1}, respectively, there are four different cases to be considered.

<u>Case 1</u>: $s_j = H_j g_j$ and $\sigma_j < \overset{*}{\sigma}_j$, i.e., the same constraints are active at x_j and x_{j+1}. Therefore, $T_{j+1} = T_j$ and we can choose

$$c_{i,j+1} = c_{ij}, \quad i = 1, \ldots, q .$$

In order to obtain superlinear convergence we determine a new basis $c_{q+1,j+1}, \ldots, c_{n,j+1}$ for T_j such that

$$H_{j+1} = \sum_{i=q+1}^{n} c_{i,j+1} c_{i,j+1}'$$

satisfies the quasi-Newton equation, i.e.,

$$H_{j+1} d_j = p_j ,$$

where

$$d_j = \frac{g_j - g_{j+1}}{\|\sigma_j s_j\|} \quad \text{and} \quad p_j = \frac{s_j}{\|s_j\|} .$$

To this end compute

$$\omega_{ij} = \frac{1}{s_j'(g_j - g_{j+1})} \left[c_{ij}' g_{j+1} - c_{ij}' g_j \left(1 - \sqrt{\left(1 - \frac{g_{j+1}' s_j}{g_j' s_j}\right) \sigma_j} \right) \right]$$

and set

(2.7) $$c_{i,j+1} = c_{ij} + \omega_{ij} s_j, \quad i = q+1, \ldots, n .$$

It can be shown (see [10]) that then

$$H_{j+1} = H_j + \frac{d_j' p_j + d_j' H_j d_j}{(d_j' p_j)^2} p_j p_j' - \frac{p_j d_j' H_j + H_j d_j p_j'}{d_j' p_j} .$$

In the unconstrained case this is the Broyden-Fletcher-Goldfarb-Shanno update formula for H_j (see e.g. [11]). Clearly, $H_{j+1}d_j = p_j$.

It remains to be shown that the vectors defined by (2.7) define a basis for T_j. Since $s_j \in T_j$ we have $c_{i,j+1} \in T_j$ for $i = q+1,\ldots,n$ and it suffices to show that the $c_{i,j+1}$, $i = q+1,\ldots,n$, are linearly independent or equivalently that H_{j+1} is positive definite on T_j. Let $\hat{d}_j$ and $\hat{g}_j$ denote the orthogonal projection of d_j and g_j onto T_j, respectively, and choose $u \in \operatorname{span}\{\hat{d}_j, \hat{g}_j\}$ such that $p_j'u = 0$ and $u \neq 0$ if $\hat{d}_j$ and $\hat{g}_j$ are linearly independent. Since $\hat{d}_j'p_j = d_j'p_j > 0$ every $x \in T_j$ can be written in the form

$$x = y + \rho\hat{d}_j + \lambda u ,$$

where $(H_j\hat{d}_j)'y = (H_j\hat{g}_j)y = 0$. Observing that $H_j\hat{d}_j = H_jd_j$ and $H_j\hat{g}_j = H_jg_j$ we have

$$x'H_{j+1}x = y'H_jy + \rho^2 d_j'p_j + \lambda^2 u'H_ju > 0 \quad \text{if} \quad x \neq 0 .$$

<u>Case 2</u>: $s_j = c_{qj}$ and $\sigma_j < \sigma_j^*$, i.e., the constraint $a_q'x \leq (b)_q$ is not active at x_{j+1}. Thus

$$T_{j+1} = \{x \mid a_i'x = 0, \quad i = 1,\ldots,q-1\} .$$

Since $c_{qj},\ldots,c_{nj}$ form a basis of T_{j+q} we can set

$$c_{i,j+1} = c_{ij}, \quad i = q, q+1,\ldots,n .$$

With

$$H_{j+1} = \sum_{i=q}^{n} c_{i,j+1}c_{i,j+1}'$$

we have

$$H_{j+1}x = H_jx \quad \text{for} \quad x \in T_j$$

because c_{qj} is orthogonal to $c_{q+1,j},\ldots,c_{nj}$. If we set

$$c_{i,j+1} = c_{ij} - \frac{c_{ij}'c_{qj}}{c_{qj}'c_{qj}}c_{qj}, \quad i = 1,\ldots,q-1$$

then these vectors are orthogonal to T_{j+1} and

$$a_k'c_{i,j+1} = 0, \; i,k = 1,\ldots,q-1, \; i \neq k, \; a_k'c_{k,j+1} = 1 .$$

Case 3: $s_j = H_j g_j$ and $\sigma_j = \sigma_j^*$, a new constraint is active at x_{j+1}. Let a_{q+1} be the gradient of this new active constraint. Then

$$T_{j+1} = \{x | a_i' x = 0, \quad i = 1,\ldots,q+1\} .$$

Since we consider H_j as an approximation to the inverse Hessian matrix of $F(x)$ on the subspace T_j we want to determine a basis $c_{q+2,j+1},\ldots,c_{n,j+1}$ of T_{j+1} such that with

$$H_{j+1} = \sum_{j=q+2}^{n} c_{i,j+1} c_{i,j+1}'$$

we have

$$H_{j+1} x = H_j x \quad \text{for} \quad x \in T_{j+1} .$$

This can be done by using a lemma given in [1] which in our notation is as follows.

Lemma 1: Let $v = H_j a_{q+1}$ and $\omega = v' a_{q+1} - c_{\nu j}' a_{q+1}$ where $\nu \in \{q+1,\ldots,n\}$. If $\omega = 0$ set

$$c_{i,j+1} = c_{ij}, \quad i = q+1,\ldots,n, \; i \neq \nu ,$$

otherwise set, for $i = q+1,\ldots,n$, $i \neq \nu$,

$$c_{i,j+1} = c_{ij} - c_{ij}' a_{q+1} \left(\frac{1 - t c_{\nu j}' a_{q+1}}{v' a_{q+1}} v - t c_{\nu j}' a_{q+1} \right)$$

where

$$t = \frac{-c_{\nu j}' a_{q+1} + \sqrt{\omega^2 + (c_{\nu j}' a_{q+1})^2}}{\omega}$$

is a solution of the equation

$$\omega t^2 + 2 c_{\nu j}' a_{q+1} t - 1 = 0 .$$

Then

i) $\quad a_{q+1}' c_{i,j+1} = 0, \quad i = q+1,\ldots,n, \quad i \neq \nu$

ii) $\quad \text{span}\{v, c_{ij}, i = q+1,\ldots,n, i \neq \nu\} = \text{span}\{c_{q+1,j},\ldots,c_{nj}\}$

iii) $\quad c_{i,j+1}' d_{h,j+1} = c_{ij}' d_{kj}, \; i,k = q+1,\ldots,n, \; i \neq \nu, \; k \neq \nu ,$

where $d_{kj}, d_{k,j+1} \in T_{j+1}$ such that $H_j d_{k,j+1} = c_{k,j+1}$ and $H_j d_{kj} = c_{kj}$.

Clearly, $c_{i,j+1}$, $i = q+1,\ldots,n$, $i \neq \nu$, form a basis for T_{j+1}. Furthermore,

$$H_{j+1}d_{k,j+1} = \sum_{\substack{i=q+1 \\ i\neq\nu}}^{n} c_{i,j+1}c'_{i,j+1}d_{k,j+1} = c_{k,j+1} = H_j d_{k,j+1}$$

which implies $H_{j+1}x = H_j x$ for $x \in T_{j+1}$.

To define the remaining columns of C_{j+1} let

$$\hat{a}_{q+1} = a_{q+1} - \sum_{i=1}^{q} (c'_{ij}a_{q+1})a_i .$$

Then $\hat{a}_{q+1}$ is the orthogonal projection of a_{q+1} onto T_j and therefore orthogonal to $a_1,\ldots,a_q$. Since $\hat{a}_{q+1}$ is also orthogonal to T_{j+1} each of the vectors

$$c_{\nu,j+1} = \frac{\hat{a}_{q+1}}{\hat{a}'_{q+1}\hat{a}_{q+1}}$$

$$c_{i,j+1} = c_{ij} - (a'_{q+1}c_{ij})c_{\nu,j+1}, \quad i = 1,\ldots,q ,$$

is orthogonal to T_{j+1} *and has the property that*

$$a'_k c_{i,j+1} = 0,\ k \neq i,\ k = 1,\ldots,q+1,\ a'_k c_{k,j+1} = 1 .$$

<u>Case 4</u>: $s = c_{qj}$ and $\sigma_j = \sigma^*_j$, i.e., instead of the constraint $a'_q x \leq (b)_q$ a new constraint, say, $a'_{q+1}x \leq (b)_{q+1}$ is active at x_{j+1}. If all $q = n$, i.e., if x_j is an extreme point of R, set

$$c_{q,j+1} = \frac{c_{qj}}{a'_{q+1}c_{qj}}$$

$$c_{i,j+1} = c_{ij} - \frac{c_{qj}}{a'_{q+1}c_{qj}}\, c_{qj}, \quad i = 1,\ldots,n,\ i \neq q .$$

If $q < n$ use the procedure of Case 3 to add the constraint $a'_{q+1}x \leq (b)_{q+1}$ to the set of active constraints. Denote the resulting matrix by $\tilde{C}_{j+1}$. Then use the method of Case 2 with C_j replaced by $\tilde{C}_{j+1}$ to drop the constraint $a'_q x \leq (b)_q$. The resulting matrix H_{j+1} has the property

$$H_{j+1}x = H_j x \quad \text{for} \quad x \in \{x \mid a_i x = 0,\ i = 1,\ldots,q+1\} .$$

3. Detailed statement of the algorithm

It is assumed that the algorithm starts with an extreme point x_0 of the feasible region R which can be obtained by solving a linear minimization problem. Let

$$a_i'x_0 = (b)_i, \quad i = 1,\dots,n$$
$$a_i'x_0 < (b)_i, \quad i = n+1,\dots,m .$$

Set

$$D' = (a_1,\dots,a_n) \text{ and } C_0 = D^{-1} = (c_{10},\dots,c_{n0}) ,$$

then the matrix C_0 has the properties described in the previous section. In addition to the matrix C_j we associate with each x_j, generated by the algorithm, a set

$$J(x_j) = \{\alpha_{1j},\dots,\alpha_{nj}\}$$

where $\alpha_{ij} \in \{0,1,\dots,m\}$. If $\alpha_{ij} = 0$, then c_{ij} *is orthogonal to the gradients of* all constraints active at x_j. If $\alpha_{ij} = k > 0$, then the constraint $a_k'x \le (b)_k$ is active at x_j and $a_k'c_{ij} = 1$. Clearly

$$\alpha_{i0} = i, \quad i = 1,\dots,n .$$

At the beginning of the jth cycle of the algorithm the following data is available: $x_j \in R$, $g_j = \nabla F(x_j)$, positive constants γ_j, γ, δ_1 and δ_2 with $\delta_1 < \delta_2 < 1$, $\delta_1 < 0.5$, $\gamma < 1$. Furthermore, the set $J(x_j)$ and the matrix C_j are given. The jth cycle of the algorithm consists of the following 3 steps.

<u>Step 1</u>: <u>Computation of the search direction s_j.</u> Compute $c_{ij}'g_j$ for all i with $\alpha_{ij} > 0$ and determine $k = k_j$ such that

$$c_{kj}'g_j \ge c_{ij}'g_j \quad \text{for all } i \text{ with } \alpha_{ij} > 0 .$$

If

$$\Big\| g_j - \sum_{\alpha_{ij}>0} (c_{ij}'g_j) a_{\alpha_{ij}} \Big\| \le \gamma_j c_{kj}'g_j$$

set

$$s_j = c_{kj} \text{ and } \gamma_{j+1} = \gamma\gamma_j ,$$

otherwise set

$$s_j = \sum_{\alpha_{ij}=0} (c'_{ij} g_j) c_{ij} \quad \text{and} \quad \gamma_{j+1} = \gamma_j .$$

If $s_j = 0$, stop; otherwise go to Step 2.

Step 2: Computation of the step size σ_j. If $a'_i s_j \geq 0$ for $i = 1,\ldots,m$ set $\sigma^*_j = \infty$, otherwise set

$$\sigma^*_j = \min\left\{ \frac{a'_i x_j - (b)_i}{a'_i s_j} \,\middle|\, \text{for all } i \text{ with } a'_i s_j < 0 \right\} .$$

Determine $\hat{\sigma}_j$ such that

$$F(x_j - \hat{\sigma}_j s_j) \leq F(x_j) - \delta_1 \hat{\sigma}_j g'_j s_j$$

and

$$(\nabla F(x_j - \hat{\sigma}_j s_j))' s_j \leq \delta_2 g'_j s_j$$

with $\hat{\sigma}_j = 1$ if possible. Set

$$\sigma_j = \min\{\sigma^*_j, \hat{\sigma}_j\} \quad \text{and} \quad x_{j+1} = x_j - \sigma_j s_j .$$

Compute g_{j+1} and go to Step 3.

Step 3: Computation of C_{j+1}.

Case 1: $s_j = \sum_{\alpha_{ij}=0} (c'_{ij} g_j) c_{ij}$ and $\sigma_j < \sigma^*_j$, (no change in the set of active constraints).

For all i with $\alpha_{ij} > 0$ set

$$c_{i,j+1} = c_{ij} .$$

For all i with $\alpha_{ij} = 0$ compute

$$\omega_{ij} = \frac{1}{s'_j (g_j - g_{j+1})} \left[c'_{ij} g_{j+1} - c'_{ij} g_j \left(1 - \sqrt{\left(1 - \frac{g'_{j+1} s_j}{g_j s_j} \right) \sigma_j} \right) \right]$$

and set

$$c_{i,j+1} = c_{ij} + \omega_{ij} s_j .$$

Let

$$C_{j+1} = (c_{1,j+1}, \ldots, c_{n,j+1}), \quad J(x_{j+1}) = J(x_j) ,$$

replace j with $j + 1$ and go to Step 1.

Case 2: $s_j = c_{kj}$ and $\sigma_j < \sigma_j^*$, (dropping an active constraint). Set

$$c_{i,j+1} = c_{ij} \quad \text{for } i = k \text{ and all } i \text{ with } \alpha_{ij} = 0 ,$$

$$c_{i,j+1} = c_{ij} - \frac{c_{ij}'c_{kj}}{c_{kj}'c_{kj}} c_{kj} \quad \text{for all } i \neq k \text{ with } \alpha_{ij} > 0 .$$

Set

$$C_{j+1} = (c_{1,j+1},\ldots,c_{n,j+1}) \text{ and } J(x_{j+1}) = \{\alpha_{1,j+1},\ldots,\alpha_{n,j+1}\}$$

where

$$\alpha_{i,j+1} = \alpha_{ij} \quad \text{for } i = 1,\ldots,n,\ i \neq k$$

$$\alpha_{k,j+1} = 0 .$$

Replace j with $j + 1$ and go to Step 1.

Case 3: $s_j = \sum_{\alpha_{ij}=0} (c_{ij}'g_j)c_{ij}$ and $\sigma_j = \sigma_j^*$, (adding a new active constraint). Let a_ℓ be the gradient of the new active constraint. Select any ν with $\alpha_{\nu j} = 0$ and compute

$$v_j = \sum_{\alpha_{ij}=0} (c_{ij}'a_\ell)c_{ij} \quad \text{and} \quad \omega_j = v_j'a_\ell - (c_{\nu j}'a_\ell)^2 .$$

If $\omega_j = 0$ set

$$c_{i,j+1} = c_{ij} \quad \text{for all } i \neq \nu \text{ with } \alpha_{ij} = 0 ,$$

otherwise

$$c_{i,j+1} = c_{ij} - c_{ij}'a_\ell\left[\frac{1 - t_j c_{\nu j}'a_\ell}{v_j'a_\ell} v_j - t_j c_{\nu j}'\right] ,$$

where

$$t_j = \frac{1}{\omega_j}\left[\sqrt{\omega_j^2 + (c_{\nu j}'a_\ell)^2} - c_{\nu j}'a_\ell\right].$$

Compute

$$\hat{a}_\ell = a_\ell - \sum_{\alpha_{ij}>0} (c_{ij}'a_\ell)a_{\alpha_{ij}}$$

and set

$$c_{\nu,j+1} = \frac{\hat{a}_\ell}{\hat{a}_\ell'\hat{a}_\ell}$$

$$c_{i,j+1} = c_{ij} - (c_{ij}'a_\ell)c_{\nu,j+1} \quad \text{for all } i \text{ with } \alpha_{ij} > 0 .$$

Set

$$C_{j+1} = (c_{1,j+1},\dots,c_{n,j+1}) \quad \text{and} \quad J(x_{j+1}) = \{\alpha_{1,j+1},\dots,\alpha_{n,j+1}\}$$

where

$$\alpha_{i,j+1} = \alpha_{ij}, \quad \text{for } i = 1,\dots,n, \; i \neq \nu$$
$$\alpha_{\nu,j+1} = \ell .$$

Replace j with $j + 1$ and go to Step 1.

Case 4: $s_j = c_{kj}$ and $\sigma_j = \sigma_j^*$, (adding and dropping an active *constraint*). *If* $\alpha_{ij} > 0$, $i = 1,\dots,n$ set

$$c_{k,j+1} = \frac{c_{kj}}{a_\ell'c_{kj}}$$

$$c_{i,j+1} = c_{ij} - \frac{a_\ell'c_{ij}}{a_\ell'c_{kj}} c_{kj}, \quad i = 1,\dots,n, \; i \neq k .$$

If at least one $\alpha_{ij} = 0$ use the procedure given in Case 3 to compute a new matrix $\tilde{C}_{j+1}$. Then use the method of Case 2 with C_j and $\alpha_{\nu j}$ replaced with $\tilde{C}_{j+1}$ and ℓ, respectively, to determine C_{j+1} and $J(x_{j+1})$. Replace j with $j + 1$ and go to Step 1.

Remark:

i) In Step 1 we set $\gamma_{j+1} = \gamma\gamma_j < \gamma_j$ whenever $s_j = c_{kj}$, i.e., whenever an active constraint is dropped. Since in the convergence proof we only use the fact that $\{\gamma_j\}$ is a convergent sequence of positive numbers with

$$\lim_{j\to\infty} \gamma_j = 0 \quad \text{iff} \quad s_j = c_{kj} \quad \text{for infinitely many } j$$

any method which produces a sequence with these properties can be used.

ii) The algorithm can easily be modified to handle linear equality constraints. Since equality constraints are always active the only difference is that a vector c_{ij}

corresponding to an equality constraint is not a candidate for the search direction s_j in Step 1 of the algorithm. This could be indicated by choosing $\alpha_{ij} = -1$ for all i such that c_{ij} corresponds to an equality constraint.

4. Superlinear convergence

First we observe that for each x_j and x_{j+1} generated by the algorithm we have $x_j, x_{j+1} \in R$ and $F(x_{j+1}) < F(x_j)$. Furthermore, the algorithm terminates with an x_j if and only if x_j satisfies the Kuhn-Tucker conditions and is therefore an optimal solution.

We assume now that the algorithm generates an infinite sequence $\{x_j\}$ and shall prove that, under the assumption stated in Section 2, this sequence converges superlinearly to the optimal solution z. The convergence proof is closely related to the proof given by Fischer [4].

Lemma 2: There is j_0 and $I \subset \{1,\ldots,m\}$ such that, for $j \geq j_0$,

$$a_i'x_j = (b)_i, \quad i \in I$$
$$a_i'x_j < (b)_i, \quad i \notin I .$$

Proof: For every j let $I_j \subset \{1,\ldots,m\}$ be such that $i \in I_j$ if and only if $a_i'x_j = (b)_i$. Furthermore, let $J \subset \{0,1,\ldots\}$ be such that an active constraint is dropped at x_j if and only if $j \in J$.

Suppose that the lemma is not true. Then J is an infinite set and

$$\Omega = \{I \subset \{1,\ldots,m\} | I = I_j \text{ for infinitely many } j \in J\}$$

is non-empty. Choose any $\tilde{I} \in \Omega$ which has the maximal number of elements of all $I \subset \Omega$. Set

$$J_1 = \{j \in J | I_j = \tilde{I}\} .$$

There is $k \subset \tilde{I}$ and an infinite subset $J_2 \subset J_1$ such that for each x_j, $j \in J_2$, always the constraint $a_k'x \leq (b)_k$ is dropped from the set of active constraints. Since $\gamma_j \to 0$ as $j \to \infty$ it follows from Step 1 of the algorithm that $\hat{g}_j \to 0$ as $j \to \infty$, $j \in J_2$, where $\hat{g}_j$ denotes the orthogonal projection of g_j onto

$$T = \{x | a_i'x = 0, \ i \in \tilde{I}\} .$$

By the uniform convexity of $F(x)$ this implies

$$x_j \to \tilde{x} \text{ as } j \to \infty, \ j \in J_2$$

where $\tilde{x} \in R$ is the unique solution of

$$\min\{F(x) \mid a_i'x = (b)_i,\ i \in \tilde{I}\}\ .$$

Let

$$\nabla F(\tilde{x}) = \sum_{i \in \tilde{I}} \lambda_i a_i \quad \text{and} \quad \lambda = \max\{\lambda_i,\ i \in \tilde{I}\}\ . \tag{4.1}$$

First suppose that $\lambda \le 0$. By Taylor's theorem there are numbers $0 \le \xi_j, \eta_j \le 1$ such that for, $j \in J_2$,

$$\begin{aligned} c_{kj}'g_j &= c_{kj}'\nabla F(\tilde{x}) + c_{kj}G(\tilde{x} + \xi_j(x_j - \tilde{x}))(x_j - \tilde{x}) \\ &\le \|c_{kj}\|\,\|G(\tilde{x} + \xi_j(x_j - \tilde{x}))\|\,\|x_j - \tilde{x}\| \le \eta\|c_{kj}\|\,\|x_j - \tilde{x}\| \end{aligned}$$

and

$$(x_j - \tilde{x})'g_j = (x_j - \tilde{x})'\nabla F(\tilde{x}) + (x_j - \tilde{x})G(\tilde{x} + \eta_j(x_j - \tilde{x}))(x_j - \tilde{x}) \ge \mu\|x_j - \tilde{x}\|^2 .$$

Thus $(x_j - \tilde{x})'\hat{g}_j = (x_j - \tilde{x})'g_j$ implies $\|\hat{g}_j\| \ge \mu\|x_j - \tilde{x}\|$, and we have

$$\|\hat{g}_j\| \ge \frac{\mu}{\eta\|c_{kj}\|}\, c_{kj}'g_j \quad \text{for} \quad j \in J_2\ .$$

Since $\gamma_j \to 0$ *as* $j \to \infty$ *we obtain the contradiction that for* $j \in J_2$ *sufficiently* large no active constraint is dropped.

To complete the proof it suffices, therefore, to show that $\lambda \le 0$. Suppose $\lambda > 0$. Since

$$c_{kj}'g_j = \max\{c_{ij}'g_j \mid \alpha_{ij} > 0\} \to \lambda \quad \text{as} \quad j \to \infty,\ j \in J_2$$

and $s_j = c_{kj}$ for $j \in J_2$, it follows that

$$g_j's_j \ge \varepsilon\|s_j\| \quad \text{for} \quad j \in J_2 \quad \text{and some} \quad \varepsilon > 0\ .$$

Since $F(x)$ is bounded from below Step 2 of the algorithm implies that

$$\sigma_j^* \to 0 \quad \text{as} \quad j \to \infty,\ j \in J_2$$

and

$$\sigma_j^* = \sigma_j \quad \text{for infinitely many} \quad j \in J_2\ .$$

Thus there is $\ell \in \{1,\ldots,m\} - \tilde{I}$ and an infinite subset $J_3 \subset J_2$ such that the constraint $a_\ell'x \le (b)_\ell$ is active at x_{j+1}, $j \in J_3$. Furthermore $a_\ell'\tilde{x} = (b)_\ell$ since $\|x_{j+1} - x_j\| \to 0$ as $j \to \infty$, $j \in J_3$.

For $j \in J_3$ let $r_j \geq 0$ be the largest integer such that for $x_{j+1}, \ldots, x_{j+r_j}$ the same constraints are active. Since the definition of $\tilde{I}$ implies that for all but at most finitely many x_{j+r_j} an active constraint is dropped it follows again from Step 1 of the algorithm that

$$x_{j+r_j} \to x^* \text{ as } j \to \infty, \; j \in J_3$$

where x^* is the unique solution of

$$\min\{F(x) \,|\, a_i'x = (b)_i, \; i \in \tilde{I} - \{k\} + \{\ell\}\} . \tag{4.2}$$

Let $I^* = \tilde{I} - \{k\} + \{\ell\}$ and

$$\nabla F(x^*) = \sum_{i \in I^*} \tau_i a_i . \tag{4.3}$$

Because $\tilde{x}$ is a feasible solution of problem (4.2) and $F(\tilde{x})$ and $F(x^*)$ are both cluster points of the monotone decreasing sequence $\{F(x_j)\}$ it follows that $F(\tilde{x}) = F(x^*)$ and $\tilde{x} = x^*$. Subtracting (4.3) from (4.2) we have therefore

$$\lambda_k a_k - \tau_\ell a_\ell = 0 ,$$

which by the linear independence of gradients of active constraints gives the contradiction that $\lambda = \lambda_k = 0$.

<u>Theorem</u>: Let Assumption 1 be satisfied. The sequence $\{x_j\}$ converges superlinearly to the optimal solution of the problem

$$\min\{F(x) \,|\, Ax \leq b\} .$$

For j sufficiently large $\sigma_j = 1$.

<u>Proof</u>: Let j_0 and I be defined as in Lemma 2. For $j \geq j_0$ the application of the algorithm to the given problem is equivalent to its application to the problem

$$\min\{F(x) \,|\, a_i'x = (b)_i, \; i \in I\} . \tag{4.4}$$

Therefore, it follows from Theorem 2.1 in [4] that $\sigma_j = 1$ for j sufficiently large and that $\{x_j\}$ converges superlinearly to the optimal solution z of (4.4). Let

$$\nabla F(z) = \sum_{i \in I} \lambda_i a_i ,$$

and let $\hat{g}_j$ denote the orthogonal projection of g_j onto $\{x \mid a'x = 0,\ i \in I\}$. Since $\gamma_j \geq \gamma^*$ for some $\gamma^* > 0$, $\|\hat{g}_j\| \to 0$ as $j \to \infty$ and

$$c'_{kj} g_j \to \lambda = \max\{\lambda_i,\ i \in I\} \quad \text{as} \quad j \to \infty\ ,$$

it follows from $\|\hat{g}_j\| > \gamma_j c'_{kj} g_j$ for $j \geq j_0$, that $\lambda \leq 0$. Since $z \in R$, it is the optimal solution of the given problem.

REFERENCES

[1] M. J. Best and K. Ritter, "An effective algorithm for quadratic minimization problems", TSR #1691, Mathematics Research Center, University of Wisconsin, 1976.

[2] K. W. Brodlie, A. R. Gourlay, and J. Greenstadt, "Rank one and two corrections to positive definite matrices expressed in product form", J. Inst. Math. Appl. 11 (1973), 73-82.

[3] W. C. Davidon, "Optimally conditioned optimization algorithms without line searches", Mathematical Programming 3 (1975), 1-30.

[4] J. Fischer, "Methoden mit variabler Metrik für Minimierungsprobleme mit linearen Nebenbedingungen", Dissertation, Universität Stuttgart 1977.

[5] R. Fletcher and M. J. D. Powell, "A rapidly convergent descent method for minimization", Computer Journal 6 (1963), 163-168.

[6] P. E. Gill and W. Murray, eds., "Numerical methods for constrained optimization", Academic Press 1974.

[7] D. Goldfarb, "Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints", SIAM J. Appl. Math. 17 (1969), 739-764.

[8] O. L. Mangasarian, "Nonlinear Programming", McGraw-Hill, 1969.

[9] M. J. D. Powell, "Some global convergence properties of a variable metric algorithm for minimization without exact line searches", Computer Science and Systems Division, Atomic Energy Research Establishment, Harwell, 1975.

[10] K. Ritter, "A product version of a variable metric method and its convergence properties", TSR #1677, Mathematics Research Center, University of Wisconsin, 1976.

[11] D. F. Shanno, "Conditioning of Quasi-Newton methods for function minimization", Mathematics of Computation 24 (1970), 647-656.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

14

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
6 MRC-TSR-1798

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
9 Technical summary rept.

4. TITLE (and Subtitle)
A VARIABLE METRIC METHOD FOR LINEARLY CONSTRAINED MINIMIZATION PROBLEMS.

5. TYPE OF REPORT & PERIOD COVERED
Summary Report - no specific reporting period

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
10 Klaus Ritter

8. CONTRACT OR GRANT NUMBER(s)
15 DAAG29-75-C-0024

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Mathematics Research Center, University of Wisconsin
610 Walnut Street
Madison, Wisconsin 53706

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
5 (Mathematical Programming and Operations Research)

11. CONTROLLING OFFICE NAME AND ADDRESS
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, North Carolina 27709

12. REPORT DATE
11 Oct [redacted] 77

13. NUMBER OF PAGES
1 12 22p.

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
linearly constrained minimization
variable metric method
superlinear convergence

x sub j - sigma sub j (s sub j) sub j sub 1j sub nj sub j sub j+1

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An algorithm is described for minimizing a nonlinear function subject to linear inequality constraints. The method generates a sequence $\{x_j\}$ with $x_{j+1} = x_j - \sigma_j s_j$, where s_j and σ_j denote the search direction and the step size, respectively. Associated with each x_j is an (n,n)-matrix $C_j = (c_{1j}, \ldots, c_{nj})$ which is used to compute s_j as a suitable linear combination of $c_{1j}, \ldots, c_{nj}$. At each iteration the matrix C_j is updated. The update formula depends on the constraints that are active at x_j and x_{j+1}, respectively. Under appropriate assumptions it is shown that $\sigma_j = 1$ for j sufficiently large and that $\{x_j\}$ converges superlinearly to the optimal solution of the minimization problem.

sub j

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

sigma sub j   sub j   sub j+1

221 200